

Library

Deadline: 13.6.2021

Points: 25

Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)



Topic description

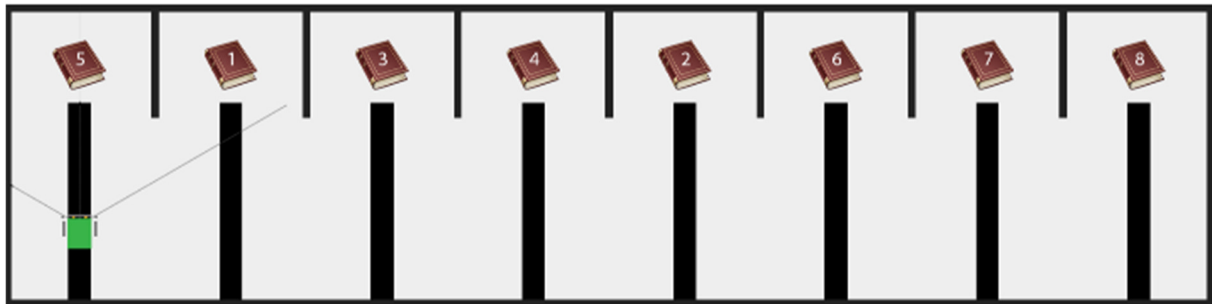
Another new assignment for you, the library in your town has asked you to develop a program for their robot, which is able to sort the books in the library. Before you start to work on a real robot, in this task you have to develop a prototype in simulation and try out different sorting algorithms.

We wish you a lot of fun with this and the following tasks!

ATTENTION: This is a fairly complex simulation task, so if this is your first time in our simulation environment, try one of the other tasks beforehand and familiarize yourself with the robot and the environment.

Task description

Visit our online simulation environment again (ide.pria.at) and create a new project with the "Library" task as a base. Open your project and the simulation task should look like this.



Two blockly files have already been created, **algo1.blockly** and **algo2.blockly**.

Your task now is to implement at least two different sorting algorithms, which bring the books into the correct order. The books themselves will be shuffled every time you start your program.

You can find out which sorting algorithms are available on the internet. **ATTENTION:** For many sorting algorithms you can find ready-to-use code on the internet, which you are welcome to use as a template, but note that many of the algorithms are implemented with recursion



Library

Deadline: 13.6.2021

Points: 25

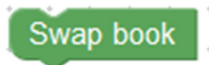
Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)

(i.e. the self-call of a function) by default. However, recursive functions can only be used with Blockly to a very limited extent, so you should always look for the equivalent iterative implementation (i.e. with the use of loops) of the respective algorithm.

Swap books

Your robot is able to pick up books that are in front of it and exchange them with the book in its memory. However, the robot can only have one book in its memory at a time.

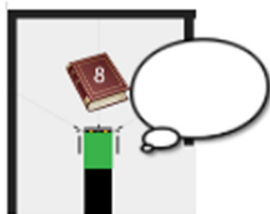
You can find the required block under "Custom" -> "Swap book"



In order to pick up or drop books, you need to touch the drop spot of a book with your front touch sensor (digital sensor 8) and then call the "Swap book" block. Here are a few examples:

Example 1

Robot memory: Empty



After swapping, the robot's memory now contains book 8 and the space in front of the robot is empty





Library

Deadline: 13.6.2021

Points: 25

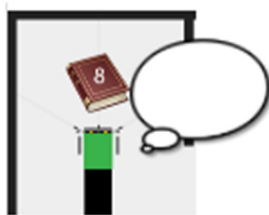
Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)

Example 2

Robot memory: Book 8

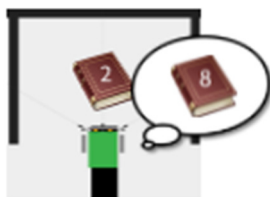


After the exchange the memory of the robot is empty and book 8 is in front of the robot

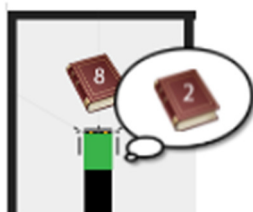


Example 3

Robot memory: Book 8



After the exchange, in the memory of the robot is the book 2 and book 8 is before the robot



Library

Deadline: 13.6.2021

Points: 25

Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)

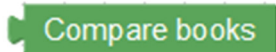
If the robot is not at one of the book locations when the swap block is called, then nothing happens.



Compare books

Your robot has the possibility to make a comparison between the book in its memory and the book in front of it.

You can find the block for this under "Custom" -> "Compare books".



The block will then return you one of the following results:

1. **Case:** Book in memory is bigger than book in front of robot -> positive number (>0) e.g. 24.
2. **Case:** book in memory is smaller than book in front of robot -> negative number (<0) e.g. -33
3. **Case:** You have no book in the memory, you are not standing at a book place, the book place you are standing at is empty -> the block returns 0

In order to compare books, you have to touch a book's shelf with your front touch sensor (digital sensor 8) and then call the " Compare books" block.

Measure Time

As described before, you have to measure the time your robot needs to sort the books. For this you have two more "Custom" blocks at your disposal.

With the block "Start stopwatch" you start the timer of a stopwatch.



With the block "Stop stopwatch and get time" you stop a previously started stopwatch and get the time in seconds that has passed since then.

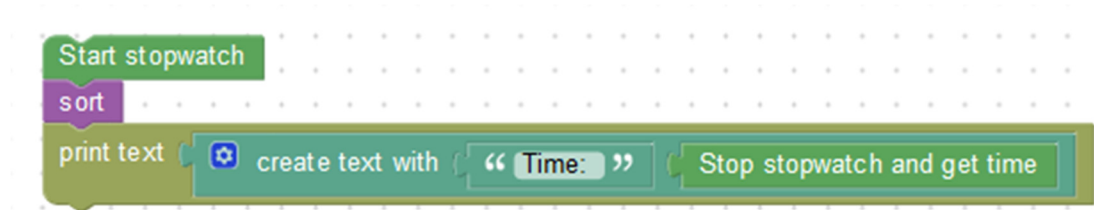
This is how your program could look like:

Library

Deadline: 13.6.2021

Points: 25

Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)



Sensors

Your robot has the following sensors available:

- Sensor 0: left outer line sensor (front left of the robot) - Analog
- Sensor 1: left inner line sensor (slightly left of center, at the front of the robot) - Analog
- Sensor 2: right inner line sensor (slightly to the right of the center, at the front of the robot) - Analog
- Sensor 3: right outer line sensor (front right of robot) - Analog
- Sensor 4: left distance sensor (about 45 degrees to the left front) - Analog
- Sensor 5: center distance sensor (straight forward) - Analog
- Sensor 6: right distance sensor (about 45 degrees to the right front) - Analog
- Sensor 8: front touch sensor (at the front of the robot) - Digital

Submission

Submit your Blockly files and a screen video of each of the implemented algorithms, as well as the document with the table of time measurements and further explanations.

Tips:

- For this task you will necessarily need functions and variables, so familiarize yourself with them.
- Write a function which you can pass the position to which the robot should move as a parameter (argument).
- Use the sensors on your robot to orientate yourself. (e.g. count the black lines to track the position of the robot).
- For many sorting algorithms you can find ready-to-use code on the internet, which you are welcome to use as a template, but note that many of the algorithms are

Library

Deadline: 13.6.2021

Points: 25

Material requirements: PC or Laptop (no Tablet) with Browser (no Safari)



implemented with recursion (i.e. the self-call of a function) by default. However, recursive functions are very limited in Blockly, so you should look for the equivalent iterative implementation (i.e. using loops) of the respective algorithm.

- If your robot does not drive straight after turns, align yourself with the walls every now and then by driving the robot against a wall, after which the robot should be straight again.

Evaluation criteria

- Robot sorts the books
- Quality of programs and documents
- Screen videos and blockly program files handed in
- Submit the document with time measurements and explanations.
- Jury evaluation: The jury evaluates the quality and creativity of the submission.